Curso básico de tecnologías XML

2. Creando documentos XML bien formados

INAP
INSTITUTO NACIONAL DE ADMINISTRACIÓN PÚBLICA

Contenido

1. Introducción.

A pesar de que XML no sea un lenguaje de programación convencional, esto no significa que no tenga cosas en común con el resto de los lenguajes de programación. Al igual que en un lenguaje de programación estructurado tipo C o Ada, podemos escribir un código que no tenga errores sintácticos y compile correctamente, con XML, podemos crear un documento con una estructura correctamente formada y adecuada a la sintaxis de XML pero que no siga las convenciones dictadas por una **DTD** (Definición de tipo de documento) o un **esquema XML** (se ven posteriormente, pero adelantaremos que es código XML que indica como debe formarse un documento XML). En ambos casos el código sería inservible. En el lenguaje de tipo C o Ada, lo es porque de nada sirve que el código compile correctamente si luego no desempeña la labor para la que ha sido creado. En el lenguaje XML, de nada sirve que creemos un documento correcto desde el punto de vista sintáctico, si no cumple las especificaciones recogidas en esa DTD o esquema.

Cuando cumple esas especificaciones y además es correcto desde el punto de vista sintáctico, le llamamos **documento XML válido**.

2. Objetivos.

En esta unidad se pretenden alcanzar los siguientes hitos:

1. Saber distinguir entre los distintos elementos que conforman la estructura de un documento XML.
2. Aprender a declarar atributos y elementos dentro de un documento XML
3. Comprender el término de documento XML 'bien formado', hecho necesario para la constitución de un documento XML funcional.

Al finalizar la unidad el alumno debe ser capaz de construir un documento XML bien formado, conociendo además los elementos que lo componen.

3. Elementos y atributos.

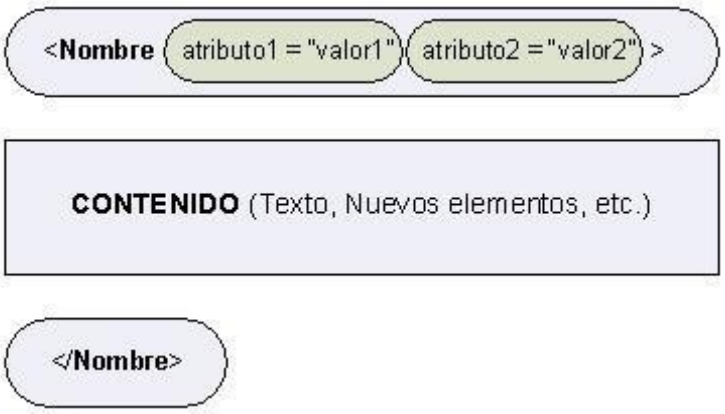
3.1 Elementos.

Se puede definir elemento en el entorno de XML como el conjunto de fragmentos de los que se compone una **etiqueta** (tag) determinada. A continuación enumeramos una serie de partes que lo componen de manera obligatoria u opcional:

- **Etiqueta de cabecera o identificador.** Define el comienzo del elemento. [*Obligatorio*]

- **Atributos**. [*Opcionales*]
- El **cuerpo o contenido**. En ocasiones, podemos declarar elementos vacíos, aunque no es una práctica muy común. [*Opcional*]
- La **etiqueta de cierre**. Define el final del elemento. [*Obligatorio*]

La estructura general de un elemento con contenido es la que muestra la siguiente figura:



Podemos crear elementos únicamente con texto como el siguiente ejemplo:

```
<Nombre>Aprender XML</Nombre>
```

O crear elementos que contengan a su vez otros elementos y texto:

```
<Nombre>Aprender XML
    <datos> Es un lenguaje muy potente </datos>
</Nombre>
```

También es posible, introducir elementos vacíos, respetando la sintaxis vista anteriormente:

```
<Texto> Ahora veremos un elemento vacío <vacio/> </Texto>
```

Los elementos no pueden superponerse. Es decir, no se puede abrir la etiqueta de un elemento, abrir la etiqueta de un nuevo elemento dentro de ese, y antes de cerrar la etiqueta de éste último, cerrar la etiqueta del primer elemento.

3.2. Atributos

Ya hemos visto en la definición de elemento, que éstos pueden contener distintos atributos. Ahora vamos a ver un poco más a fondo en que consisten. **Los atributos son propiedades que nos ofrecen una definición más concreta del elemento**. Podemos considerar los atributos como pares Nombre-Valor que se asocian a un elemento determinado. Ese nombre y su valor correspondiente son de tipo string. Un elemento en principio, puede contener un número ilimitado de atributos.

Recordemos a continuación la forma de declararlos:

Existen los siguientes nombres de atributo reservados para propósitos especiales. Estos atributos se identifican fácilmente porque llevan el prefijo '**xml**'. Veamos cuáles son:

Xml:lang

Se utiliza para clasificar un contenido por el idioma en que está escrito. Su forma de utilización es la siguiente:

Xml:lang="código que identifica el idioma en el que escribimos el contenido"

Por ejemplo, para español se pone el código '*es*', para inglés se pone '*en*', etc.

Xml:Space

Se utiliza para que un contenido mantenga la representación de los espacios en blanco. Este atributo especial, toma los valores '*preserve*' o '*default*'.

El primero de los valores se usa para conservar esos espacios en blanco. Si usamos la opción *default*, se utilizará el tratamiento por defecto.

Xml:Link

Este tipo de atributo se utiliza para el procesador *XLink*. Su uso se limita a indicar que el elemento al que pertenece el atributo es un enlace a otro elemento.

Xml:Attribute

Se utiliza para prevenir conflictos con otros posibles atributos.

Hemos visto hasta ahora que XML es un metalenguaje bastante potente para tareas como la organización de la información, debido a la creación de nuestros propios elementos. Como sucede a la hora de trabajar en HTML, es posible que en ocasiones tengamos dudas acerca de cuándo crear un nuevo elemento y de cuando incluir un atributo. Es una situación que deberá resolver el programador de XML, porque no es

una cuestión de fácil respuesta. Hay quién considera que un atributo no es más que un tipo de metadato. Es una cuestión similar a la que se plantea cuando diseñamos una base de datos.

3.2.1. ¿Cuándo un dato es necesario que sea una entidad y cuando puede ser suficiente declararlo como atributo de otra entidad?

En el caso de XML, podemos argumentar que el hecho de crear un elemento hijo en lugar de un atributo puede darnos mayor versatilidad a la hora de ampliar el código, pero quizás haya quién piense que no es razón de peso suficiente.

A continuación daremos algunas premisas que pueden ayudarnos a decidir en algunos casos:

- Cuando el dato contiene subestructuras se declarará como *elemento*.
- Cuando el dato tiene un tamaño considerable de información se declarará como *elemento*.
- Cuando el dato puede cambiar con facilidad se declara como *elemento*.
- Cuando el dato va a ser utilizado por una aplicación se declara como *elemento*.
- Cuando el dato es pequeño y cambia pocas veces de declara como *atributo*.
- Cuando el dato sólo tiene unos pocos valores fijos se declara como *atributo*.

Por último mostraremos un ejemplo de un extracto de documento XML en el que se declaran algunos atributos:

```
<?xml version="1.0"?>
<Partido jornada="7ª" dia="17/09/2006">
    España-Rumanía <resultado>2-2</resultado>
    <jugadores titulares="si">
      Iker Casillas
      Michel Salgado
      Sergio Ramos
      . . .
    </jugadores>
</Partido>
```

Los atributos han sido resaltados en negrita.

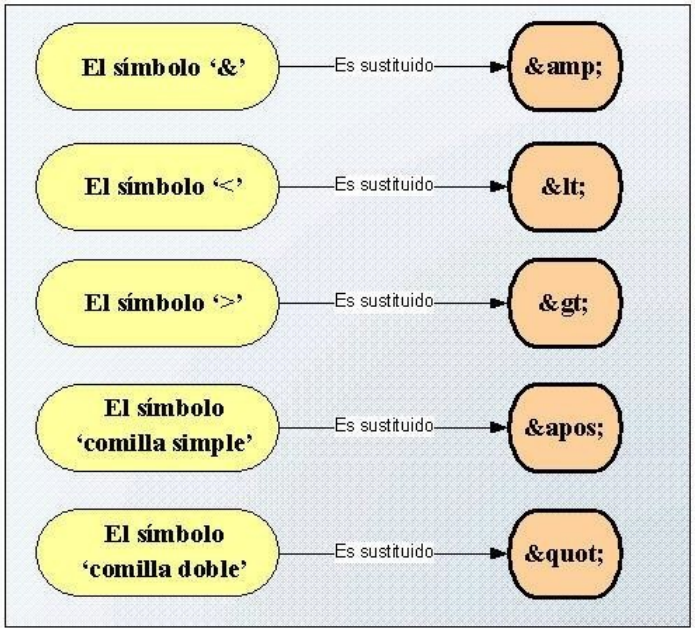
4. Referencias a Entidades.

En ocasiones, podemos encontrarnos con algún problema a la hora de incluir algún símbolo en el contenido de un elemento. Esto sucede por ejemplo, con el signo de me-

nor que (‘<’). XML y HTML interpretan estos símbolos como el comienzo de una etiqueta y por lo tanto, hasta que no aparezca un símbolo de mayor que (‘>’) no consideran que dicha etiqueta está finalizada.

Debido a que es muy normal incluir elementos o etiquetas dentro de otros elementos, XML utiliza notaciones especiales para determinados símbolos, de forma que estos sean interpretados correctamente. Estas notaciones especiales se conocen con el nombre de **entidades predefinidas** y siempre van situadas entre los caracteres *ampersand* (&) y *punto y coma* (;).

Las entidades predefinidas que tiene XML son las que se muestran en la siguiente figura



Para ilustrar esta situación, podemos ver algunos ejemplos:

```
<?xml version="1.0" encoding="UTF-8"?>
<entidades_predefinidas>
<funcionamiento>El funcionamiento de las entidades
predefinidas se puede comprobar en este ejemplo,
en el que se construye una serie de
etiquetas emulando a un fichero HTML</funcionamiento>
   <ejemplo>
     &lt;HEAD&gt;
       &quot;ejemplo&quot;
     &lt;/HEAD&gt;
   </ejemplo>
</entidades_predefinidas>
```

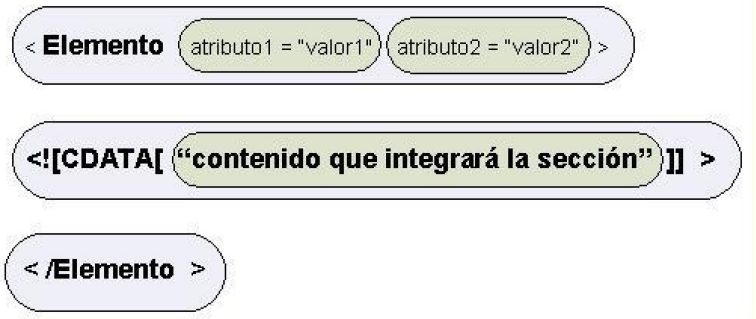
```
<Operacion> La operación  25 < 30 no sería bien interpretada </Operacion>
<Operacion> La operación  25 &lt; 30 será bien interpretada </Operacion>
```

Como se puede ver en los ejemplos, su utilización no reviste ninguna dificultad y su uso es en muchas ocasiones imprescindible para estructurar ciertos tipos de contenido.

5. Uso de las secciones CDATA.

Las **secciones CDATA** se utilizan en XML para integrar texto o contenido que contiene caracteres susceptibles de ser mal interpretados (‘<’, ‘>’, etc.). Dichos caracteres serían considerados por el procesador o parser de XML como marcas del lenguaje. La pregunta que nos podemos hacer ahora, es por qué no utilizamos las entidades predefinidas explicadas anteriormente y la respuesta es sencilla. Si queremos introducir un bloque de texto con una extensión considerable y en ese texto hay caracteres que pueden confundirse con marcas, resulta más efectivo introducir una sección de este tipo que utilizar las entidades predefinidas (estás quedan reservadas a un uso más puntual).

La sintaxis de declaración de las **secciones CDATA** es la siguiente:



Para comprender correctamente el uso de estas secciones, mostraremos un extracto de un documento XML que integra una **sección CDATA**.

```
<?xml version="1.0" encoding="UTF-8" standalone='yes'?>
<COLECCION>
  <ALBUM>
       <FOTOGRAFIA>
       <Lugar_Fotografia>Berlín(Alemania)</Lugar_Fotografia>
              <Fecha_Creacion>19/02/2005
              <Hora>
                   <![CDATA[
                         <HTML>
                            <HEAD>
                              <TITLE>Viaje Alemania</TITLE>
                            </HEAD>
                   ]]>
             </Hora>
              </Fecha_Creacion>
              <Tipo_Camara>Digital-Nikon Coolpix 4300</Tipo_Camara>
       </FOTOGRAFIA>
  </ALBUM>
</COLECCION>
```

En el ejemplo podemos ver la **sección CDATA** marcada en negrita. Si nos fijamos, dicha sección esta justificada por el uso de símbolos '<' y '>' para utilizar etiquetas del lenguaje HTML.

Gracias a esta **sección CDATA** no tenemos que preocuparnos de poner distintas entidades predefinidas.

Cualquier fragmento de texto que incluyamos en una **sección CDATA**, independientemente de la longitud de la que conste, será excluido del procesador de XML.

6. Codificación de Documentos XML.

XML ofrece la posibilidad de usar una **codificación** distinta para sus caracteres por cada entidad externa procesada en el documento XML. Actualmente todos los procesadores que hay en el mercado tienen capacidad para leer entidades en los dos formatos básicos de codificación que más se utilizan; los formatos *UTF-8* y *UTF-16*. Para ello, cuando una entidad está codificada en UTF-16 debe comenzar con la marca descrita en el Anexo E de ISO/IEC 10646 y Unicode apéndice B.

Con esta marca específica un documento XML podrá distinguir adecuadamente cuando un documento está codificado en UTF-8 y cuando un documento lo está en UTF-16.

Ya se ha hablado de este tema anteriormente, pero recordaremos que los procesadores de XML pueden admitir diversos tipos de codificación a parte de los comentados UTF-8 y UTF-16. Este tipo de entidades empezarán con un identificador del tipo de co-

dificación empleada para advertir de ello al procesador del lenguaje (se incluye en la línea que abre el documento XML).

Se recomienda que los caracteres de codificación sean registrados como charsets en la **Internet Assigned Numbers Authority** (http://www.iana.org/). Cuando queremos emplear una codificación como las que recoge dicho organismo, ese tipo de codificación debe ser referido usando sus nombres registrados. Es importante no usar mayúsculas cuando en dicho registro de la *IANA* no figure así. Puede dar problemas a la hora de fijar la codificación del documento XML.

En caso de encontrar un documento codificado de una forma no predeterminada se puede producir un error con un resultado inesperado, produciendo la inestabilidad del código. A continuación se muestra cómo se declara el tipo de codificación en un documento XML. Esto debe realizarse en la cabecera de dicho documento (en la primera línea en la que se indica que el documento contiene código XML).

```
<?xml encoding="ISO-10646-UCS-2"?>
<?xml encoding="UTF-16"?>
<?xml encoding="ISO-8859-1 a -9"?>
```

7. Conclusión.

En el desarrollo de esta unidad didáctica el alumno ha debido fijar el concepto de **documento bien formado** para la tecnología XML. Para ello, se ha visto como se estructura correctamente un documento XML (en base a una serie de elementos que pueden contener o no atributos) y algunas técnicas como el uso de **secciones CDATA**. Gracias a estas secciones y a las entidades predefinidas, el alumno debe estar preparado para elaborar un documento XML con cualquier tipo de contenido. También se ha visto cuál es el formato de codificación estándar de documentos así como el conjunto de codificaciones que soporta el procesador de código XML.

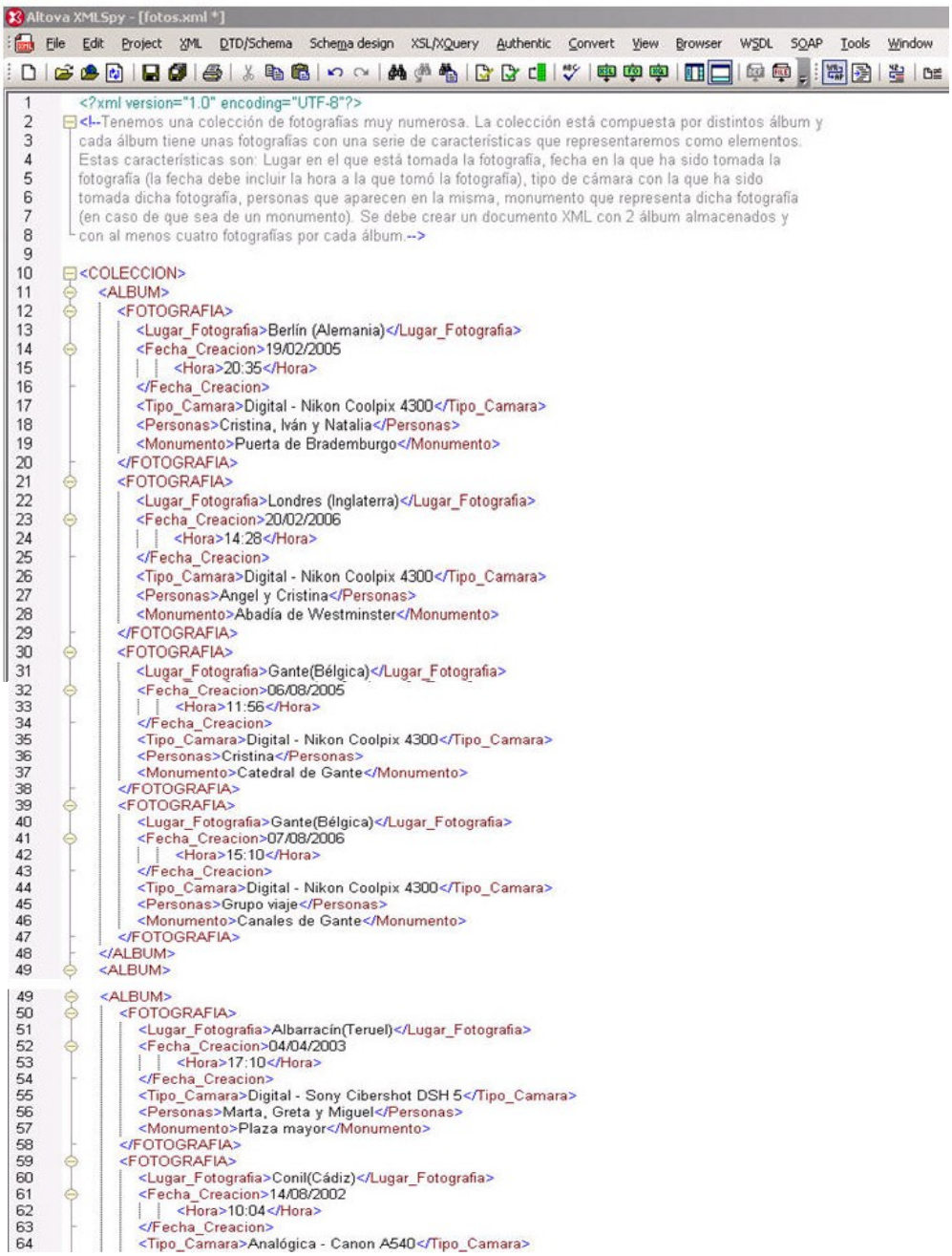
8. Ejercicio resuelto.

Se propone la creación de un documento XML que represente el siguiente dominio de información con las reglas de validez expuestas (sintaxis de etiquetas, cierre de las mismas, etc.).

DOMINIO: Tenemos una colección de fotografías muy numerosa. La colección está compuesta por distintos álbum y cada álbum tiene unas fotografías con una serie de características que representaremos como elementos. Estas características son:

Lugar en el que está tomada la fotografía, fecha en la que ha sido tomada la fotografía (la fecha debe incluir la hora a la que tomó la fotografía), tipo de cámara con la que ha sido tomada dicha fotografía, personas que aparecen en la misma, monumento que representa dicha fotografía (en caso de que sea de un monumento). Se debe crear un documento XML con 2 álbum almacenados y con al menos cuatro fotografías por cada álbum.

POSIBLE SOLUCIÓN: Teniendo en cuenta las características del domino que se pide, una posible solución puede ser la siguiente:




```
<?xml version="1.0" encoding="UTF-8"?>
<!--Tenemos una colección de fotografías muy numerosa. La colección está compuesta por distintos álbum y
cada álbum tiene unas fotografías con una serie de características que representaremos como elementos.
Estas características son: Lugar en el que está tomada la fotografía, fecha en la que ha sido tomada la
fotografía (la fecha debe incluir la hora a la que tomó la fotografía), tipo de cámara con la que ha sido
tomada dicha fotografía, personas que aparecen en la misma, monumento que representa dicha fotografía
(en caso de que sea de un monumento). Se debe crear un documento XML con 2 álbum almacenados y
con al menos cuatro fotografías por cada álbum.-->

<COLECCION>
  <ALBUM>
    <FOTOGRAFIA>
      <Lugar_Fotografia>Berlín (Alemania)</Lugar_Fotografia>
      <Fecha_Creacion>19/02/2005
          <Hora>20:35</Hora>
      </Fecha_Creacion>
      <Tipo_Camara>Digital - Nikon Coolpix 4300</Tipo_Camara>
      <Personas>Cristina, Iván y Natalia</Personas>
      <Monumento>Puerta de Brademburgo</Monumento>
    </FOTOGRAFIA>
    <FOTOGRAFIA>
      <Lugar_Fotografia>Londres (Inglaterra)</Lugar_Fotografia>
      <Fecha_Creacion>20/02/2006
          <Hora>14:28</Hora>
      </Fecha_Creacion>
      <Tipo_Camara>Digital - Nikon Coolpix 4300</Tipo_Camara>
      <Personas>Angel y Cristina</Personas>
      <Monumento>Abadía de Westminster</Monumento>
    </FOTOGRAFIA>
    <FOTOGRAFIA>
      <Lugar_Fotografia>Gante(Bélgica)</Lugar_Fotografia>
      <Fecha_Creacion>06/08/2005
          <Hora>11:56</Hora>
      </Fecha_Creacion>
      <Tipo_Camara>Digital - Nikon Coolpix 4300</Tipo_Camara>
      <Personas>Cristina</Personas>
      <Monumento>Catedral de Gante</Monumento>
    </FOTOGRAFIA>
    <FOTOGRAFIA>
      <Lugar_Fotografia>Gante(Bélgica)</Lugar_Fotografia>
      <Fecha_Creacion>07/08/2006
          <Hora>15:10</Hora>
      </Fecha_Creacion>
      <Tipo_Camara>Digital - Nikon Coolpix 4300</Tipo_Camara>
      <Personas>Grupo viaje</Personas>
      <Monumento>Canales de Gante</Monumento>
    </FOTOGRAFIA>
  </ALBUM>
  <ALBUM>
  <ALBUM>
    <FOTOGRAFIA>
      <Lugar_Fotografia>Albarracín(Teruel)</Lugar_Fotografia>
      <Fecha_Creacion>04/04/2003
          <Hora>17:10</Hora>
      </Fecha_Creacion>
      <Tipo_Camara>Digital - Sony Cibershot DSH 5</Tipo_Camara>
      <Personas>Marta, Greta y Miguel</Personas>
      <Monumento>Plaza mayor</Monumento>
    </FOTOGRAFIA>
    <FOTOGRAFIA>
      <Lugar_Fotografia>Conil(Cádiz)</Lugar_Fotografia>
      <Fecha_Creacion>14/08/2002
          <Hora>10:04</Hora>
      </Fecha_Creacion>
      <Tipo_Camara>Analógica - Canon A540</Tipo_Camara>
```

```
64          <Tipo_Camara>Analógica - Canon A540</Tipo_Camara>
65          <Personas>Victor y Javier</Personas>
66          <Monumento>Playa</Monumento>
67        </FOTOGRAFIA>
68        <FOTOGRAFIA>
69          <Lugar_Fotografia>San Sebastián(Guipuzcoa)</Lugar_Fotografia>
70          <Fecha_Creacion>09/09/1996
71            <Hora>08:56</Hora>
72          </Fecha_Creacion>
73          <Tipo_Camara>Analógica - Canon A540</Tipo_Camara>
74          <Personas>Ruth y Pedro</Personas>
75          <Monumento>Playa de la Concha</Monumento>
76        </FOTOGRAFIA>
77        <FOTOGRAFIA>
78          <Lugar_Fotografia>Llanes (Asturias)</Lugar_Fotografia>
79          <Fecha_Creacion>18/02/2003
80            <Hora>13:45</Hora>
81          </Fecha_Creacion>
82          <Tipo_Camara>Analógica - Canon A540</Tipo_Camara>
83          <Personas>Angel y Cristina</Personas>
84          <Monumento>Puerto marítimo</Monumento>
85        </FOTOGRAFIA>
86        <FOTOGRAFIA>
87          <Lugar_Fotografia>Bilbao</Lugar_Fotografia>
88          <Fecha_Creacion>15/12/2004
89            <Hora>19:26</Hora>
90          </Fecha_Creacion>
91          <Tipo_Camara>Analógica - Canon A540</Tipo_Camara>
92          <Personas>Manuel y Conchi</Personas>
93          <Monumento>Museo Guggenheim</Monumento>
94        </FOTOGRAFIA>
95        <FOTOGRAFIA>
96          <Lugar_Fotografia>Madrid</Lugar_Fotografia>
97          <Fecha_Creacion>14/04/2005
98            <Hora>22:35</Hora>
99          </Fecha_Creacion>
100         <Tipo_Camara>Digital - Nikon D50</Tipo_Camara>
101         <Personas>Ninguna</Personas>
102         <Monumento>Palacio de Correos</Monumento>
103       </FOTOGRAFIA>
104     </ALBUM>
105   </COLECCION>
106
```

9. Ejercicio propuesto.

Se propone realizar un **documento XML bien formado** que de manera similar a como se ha realizado en el ejercicio resuelto, represente un dominio relacionado con las bandas sonoras originales. El documento deberá estructurar la información de una colección de música de cine que tendrá una serie de discos y en los que se almacenará información relativa a la película a la que pertenece, el compositor que ha compuesto dicha banda sonora, el año de composición, los premios que ha obtenido la misma y el formato de compresión que tiene el disco (ej: mp3 a 192 Kbps).

Además se incluirá una sección CDATA con contenido susceptible de ser mal interpretado por el procesador de XML.